



InterActive 2

Written By: Steven Robert Cypherd

SUMMARY

This is an explanation of the code for the flow control I came up with it to get my Basic Stamp 2(BS2) talking to my Basic Stamp 1(BS1). I am showing the main loops only. For the code files look at letsmakerobots.com and instructables.com. Files or a zip file.

In my code I use a lot of Bit flags. Most processors allow access to the Bits of a Byte or Word to use like regular variables. Basic Stamp 1 and most Picaxe processors have defined variable names for the Bits of first few bytes of memory. Usually it is Bit0 – Bit7 or Bit0 – B15 or Bit0 – Bit31 to define a variable. In both of these processors the variable Bits can also be in another variable. Like my lights Byte B0 the Bits are the variables Bit0 – 7 that are lit1 – lit8 the individual lights. This makes testing of a light easy. “If lit1 = 1 then litOn” is simple.

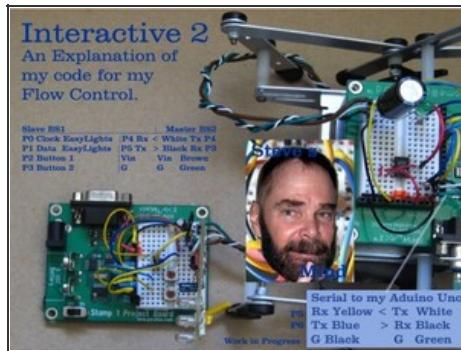
The Basic Stamp 2 and processors like it use defined subscripting to access the Bits, Nibbles and Bytes of a variable. My individual lights are the variables "lights.Bit0 - Bit7". In testing a variable Bit is “if lights.Bit0 = 1 then litOn”. Setting is “litTmp = lights.Bit0”. Usually the subscripts can not be a variable. Look at the program files in the EasyLights article. Read your manual.

I fixed as much as I could. The BS1 doesn't do serial out very well. It is stable and runs like any simple button on a system like this. The BS1 doesn't do Bits very well either.

My flow control is the only way I got these systems to communicate.

I uploaded full code pictures.

Step 1 – InterActive 2



- Get them talking. Fixed it.
 - Here is the last working version with the BS1. Two bytes in OK, but not one good byte out.
 - Here are the code pictures.

Step 2

- Last page of BS2 I2
 - BS1 I2 code.

Step 3

- First working version BS2 I1
 - First working version BS1 I1

Step 4

```

run2:Look for message from the master
IF slSel = 0 THEN jp1Set Up
IF pRx = 1 THEN jp1Check for message
SERIN pinRx, T2400, mCmd, tmp1
PAUSE 20
IF mCmd <> BS1P1 THEN jp1 'Check for correct address
lights = tmp1
GOSUB doLights
jp1:Clear buttons
IF pBln1 = 1 THEN jp2
IF pBln2 = 1 THEN jp2
bDown = 0
bIn1 = 0
bIn2 = 0
jp2:Set up slave
IF slSel = 1 THEN jp3
tmp1 = tmp1 + 1
IF tmp1 < 10 THEN jp3
slSel = 1
jp3:
GOTO runIt
END

'SubRoutines ****
doLights: From Easylights article
FOR tmp1 = 0 TO 7
    lOfftmp = tmp1
    GOSUB checkLights
    IF lOfftmp = 1 THEN doLitOn 'A simple test
    pData = lOff : GOTO doLitCk
    doLitOn:
    pData = lOn
    doLitCk:
    pClock = lOn
    pClock = lOff
    NEXT
RETURN

```

```

Each List is 1 or 0 off
checklights: 'An Array for anything
BRANCH liTmp, [clkL1,clkL2,clkL3,clkL4,clkL5,clkL6,clkL7,clkL8

clkL1; liTmp = #11: GOTO doneCheckList
clkL2; liTmp = #12: GOTO doneCheckList
clkL3; liTmp = #13: GOTO doneCheckList
clkL4; liTmp = #14: GOTO doneCheckList
clkL5; liTmp = #15: GOTO doneCheckList
clkL6; liTmp = #16: GOTO doneCheckList
clkL7; liTmp = #17: GOTO doneCheckList
clkL8; liTmp = #18

doneCheckList:
RETURN

```

- Last page BS1 If

Get them Talking.

This document was last generated on 2012-11-02 04:41:11 AM.